Jesus Ugarte

Algorithms for Machine Learning

University of Central Florida

Midterm Exam

1) Describe, in your own words, how a decision tree can be (1) built and (2) used for prediction.

A decision tree can be described as supervised learning method used for classification and regression problems and also part of the information gain learning. Decision tree models can handle both categorical and numerical data.

Diagram

Description automatically generated

Example extracted from google images

Also, this algorithm creates a prediction model that predicts the value of a target variable by just understanding a correlation from the features of the data. Hence, the deeper the tree, the more complex the decision rules and the fitter the model. Taking in consideration that fitting is a measure of how well the data is generalized to other data that may be pretty similar. Even though that there are many different classifiers and taking in consideration the problem is a classification problem, by using sklearn we would be able to import the Decision Tree Classifier which so that the Decision tree can be built by taking the input of two arrays. I will provide a short example of how the code would look like:

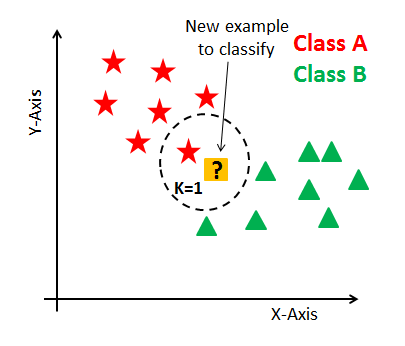
from sklearn.tree import DecisionTreeClassifier

It is necessary to note that x corresponds to the value for number of samples and y for the number of features of a given dataset. After that, now that the data Is already generalized. I would be able to go ahead and proceed with the prediction. We could use the predict or predict proba. In the first assignment I used predict\_proba since I wanted to obtain the probability value of an instance of being 1 or 0.

Decisiontree.predict\_proba(Y)

2) Describe, in your own words, how a nearest neighbors’ algorithm can be (1) built, and (2) used for prediction.

The nearest neighbor’s algorithm classifier that given an input, chooses the class of the nearest data point to that compared input. Hence, it can be inferred that the way this algorithms works is by taking a look at the nearest data point in the training stage. In terms of data handling, normalization is needed from nearest neighbors classifiers because the distance between data points is being measured.



**from** **sklearn.neighbors** **import** KNeighborsClassifier

It is important to note that considering different neighbors may give us different prediction. I will include a distribution of accuracy on the k-nearest-neighbor classifier depending on the number of neighbors that I included in my code for assignment 2:

Chart

Description automatically generated

The K-nearest-neighbor classification algorithm chooses the most common class out of the k nearest data points to that input given. Hence, if looking at the graph we can see how the accuracy varies depending of the value of K.

The way each instance for the distribution values is given by the distance between the data point and the test data for later finding the probability

In the assignment 2, Hyperparameter Search was definitely needed because the right number of neighbors, and the type of distance was needed in order to reach the benchmark score which was hard for me. I utilized a search parameter of n\_neighbors, from a range of 0 to 100 and the weight of the distance. I then Applied grid search cross validation to find the best prediction model. After that, the method .predict() could be used I order to get the predicitons.

3) Describe, in your own words, how the Naive Bayes algorithm can be (1) built and (2) used for prediction.

The Naive Bayes algorithm is an algorithm that can analyze the probability that categorizes the prediction. For the Naive bayes, It can be said the base rule of probability is the probability of b given a is the probability of a given b multiplied by the probability of b divided by the probability of a. The formula can be described as follows:

Since Naïve Bayes is a probability-based algorithm, the amount of data is not very critical for the operation of the algorithm, the training of the algorithm can be performed with a small dataset to estimate the parameters needed for classification. It is important to keep in mind that for Naïve Bayes, the probability is being weighted by both the prior and the current evidence that is available depending on the strength of the current belief. In the assignment for tumor prediction I used Gaussian NB from the sklearn library to implement my approach.

**from** **sklearn.naive\_bayes** **import** GaussianNB

I ended up using gaussian Naïve Bayes for assignment 3 and later found out that Gaussian Naïve is a variant of Naïve Models that follows a normal distribution. And, in a single attempt to obtain a simple model with no parametrization, I was able to reach to the benchmark score very easily.

4) Describe, in your own words, how an Ordinary Least Squares Regression model can be (1) built and (2) used for prediction.

The ordinary least squares regression model can be often called as linear regression. It can be described a supervised learning task of learning a function estimating an input point to a continuous value. One of the easiest ways I could think of how to imagine this would be by imagining a simple line in the Cartesian plane.

Chart, scatter chart

Description automatically generated

It can be used as an approach to define the relationship between x and y. Also, this model can be used to estimate the unknown parameters by creating a model which will minimize the sum of the squared errors between the predicted data and the observed data. According to my research this algorithm is used widely in excel to look for estimations.

We already know that the general line equation straight from college algebra

Where m is the slope, x is the independent variables and b I the intercept. On the other hand, to use the ordinary least squares regression model, we have to apply the formula given below

Text

Description automatically generated

Where x represents the independent variables, x hat represents the average of independent variables, y represents the dependent variables, and y hat represents the average of dependent variables. This algorithm should be implemented when there is an assumption of multicollinearity between any two independent variables. In pyhton, the way to implement this algorithm is as follows.

Import statsmodels.api as sn

Model = sm.OLS(b, a).fit()

Model\_prediciton = model.predict(a)

From the statsmodels api I will be able to import the necessary documentation to implement the algorithm. Both a and b array values are considered as independent features.